NOTE: Use these buttons to print or save the form. DO NOT use the browser tool bar. SAVE PRINT

LA-UR-10-07017

Approved for public release;
distribution is unlimited.

Title: Verification of the Scientific Simulations via Hypothesis-Driven Comparative and Quantitative Visualization

Author(s): James P. Ahrens 113788 CCS-7, Katrin Heitmann 175878 ISR-1, Mark R. Petersen 204932 CCS-2, Jonathan Woodring 209118 CCS-7, Sean J. Williams 230347 CCS-7, Patricia K. Fasel 090207 CCS-3, Christine M. Ahrens 113787 HPC-1, Chung-Hsing Hsu Oak Ridge National Laboratory, Berk Geveci Kitware Inc.

Intended for: IEEE Computer Graphics and Applications

Los Alamos NATIONAL LABORATORY EST.1943

Form 836 (7/06)

SAVE PRINT CLEAR FORM

Verification of Scientific Simulations via Hypothesis-Driven Comparative and Quantitative Visualization

James P. Ahrens, Katrin Heitmann,
Mark Petersen, Jonathan Woodring,
Sean Williams, Patricia Fasel, and Christine Ahrens *

Chung-Hsing Hsu †

Berk Geveci ‡

1 Keywords

Visualization in Earth, Space, and Environmental Sciences; Hypothesis Testing, Visual Evidence; Feature Detection and Tracking.

2 Abstract

We describe a visualization-assisted process for the verification of scientific simulation codes. The need for code verification stems from the requirement for very accurate predictions to interpret data confidently. We compare different cosmological and oceanographic simulations to reliably predict differences in simulation results. Our verification consists of the integration of an iterative hypothesis-verification process with comparative, feature, and quantitative visualization. We validate this process by verifying the results of different cosmology and oceanographic simulation codes.

3 Introduction

Understanding and describing the natural world, such as the role of dark energy and dark matter in the Universe or the ocean influence on the global climate, requires the use of supercomputer simulations and models. These simulations are required to be very accurate to be able to reliably interpret observational data or to make accurate predictions. A large component of simulation accuracy and the science being performed with the simulations is understanding the differences in models and algorithms in the simulations, and being able to quantify those differences.

Understanding the physics of dark energy and dark matter is the foremost challenge in cosmology today. This task requires very accurate predictions from simulation codes to interpret the observational data from ground- and

*James P. Ahrens, Katrin Heitmann, Mark Petersen, Jonathan Woodring, Sean Williams, Patricia Fasel, and Christine Ahrens are with Los Alamos National Laboratory; email: ahrens, heitmann, mpetersen, woodring, seanw, pkf, cahrens@lanl.gov.

†Chung-Hsing Hsu is with Oak Ridge National Laboratory; email: hsuc@ornl.gov.

‡Berk Geveci is with Kitware, Inc.; email: berk.geveci@kitware.com.

space-based cosmology missions. Cosmological structure formation is very sensitive to the physics of dark energy. Current observations constrain the nature of dark energy at the 10% level.

To go the next level of accuracy, observations and modeling have to improve by an order of magnitude. To achieve this goal with respect to the modeling, extensive code verification exercises must be carried out. Code verification in this context means that the accuracy of the algorithm used to solve the set of equations underlying the physical model is understood and controlled. Often this is achieved by numerically solving problems with a known analytic solution, and evaluating how well the simulation handles the problem. Though, the problem with this is that cosmological simulations are highly nonlinear, and while solvable solutions give some hints about the accuracy of the simulations, they do not capture the entirety of the problems to be solved. Therefore, to characterize the accuracy, an important part of the code verification process is the comparison of different simulations to describe the differences between algorithms.

Although people commonly consider weather to be an atmospheric phenomenon, ocean currents strongly influence regional climates. For example, the Gulf Stream carries warm water from the Gulf of Mexico to northern Europe, and is responsible for its comfortable climate (London is further north than Quebec). The El Nino oscillation, which influences droughts and floods across the globe, is the result of dynamics in a strongly coupled ocean-atmosphere system. As the climate warms, the circulation of the atmosphere and ocean will also change as some ocean currents may weaken or change course. To assess the impacts of climate change in the 21st century and beyond, scientists must accurately simulate the ocean and its interactions with the atmosphere.

Ocean models play an important role in Global Climate Models (GCMs), which are used to understand the effects of increasing greenhouse gasses on the earth's climate. Simulations based on various future emissions scenarios show how temperatures and precipitation will change over the next century. The Intergovernmental Panel on Climate Change (IPCC) incorporates climate change simulations from over twenty GCMs, by research groups around the world, into their consensus documents on climate change [12], which are used to inform policy makers.

Due to the scientifically and politically sensitive nature of climate change research, codes must be extremely well vetted. This process involves the use of idealized test cases where analytical solutions are available, convergence studies, and comparison of realistically forced high-resolution simulations to observed climatology. This also includes the process of code verification by comparison with a well-established ocean model under nearly identical conditions.

To complete the task of simulation code verification, we integrate *iterative hypothesis-verification* with *comparative, feature, and quantitative visualization*. The first step in our process is identifying a measurable feature from the output of the simulations that we wish to test or compare. Next, we formulate a hypothesis about the feature that we wish to test. Following that, we generate a qualitative, visual comparison of the different simulation results to gain intuition of the differences or similarities between the codes. Finally, we perform a quantitative, numerical comparative visualization to accurately test the hypothesis. If necessary, we repeat the process, starting at the beginning, until we are satisfied with the results.

Our iterative hypothesize and visualize process allows us or any scientist to gain a deeper understanding of the differences and similarities between simulations. Furthermore, this process fits well with the scientific method, through generation from a step-by-step document of evidence for reproducibility, a critical component for science. Additionally, we believe this elucidation of a specific visualization process for a specific goal, in this case scientific code verification, is critical to describing and advancing the science of visualization. As a part of that, our method and the case studies we describe in the following ideally should be part of a "visualization cookbook": a list of the tried and true task specific visualization methods and processes. The metrics for our success are the reproducibility of our method and the ease of completion when we apply our process to real-world code verification problems.

4 Related Work

We classify related approaches to our code verification process into three categories: iterative hypothesis-verification, comparative visualization, and feature extraction and quantitative analysis.

In hypothesis verification, Chen et al. [1] describe the visualization process as an iterative search process. Each iteration produces a visualization that may increase your information and knowledge. In the general case, the search space is extremely large. For example, the parameter configurations, such as types of visualization algorithms and view positions, are just a few of the search space axes that can be considered.

Kehrer et al. [7] explore this search space by rapidly generating hypotheses. They identify regions in climate data which are sensitive to atmospheric climate change, and then statistically verify whether or not these regions are robust indicators of climate change. Iterative visualization and interaction is used with sensitivity metrics to narrow down the regions of interest. We differ in our approach by narrowing the problem search space to code verification and providing a structured process for traversing the space. Our iterative process builds up a body of evidence that highlights the differences between the simulations.

To understand the sources of inconsistency between different simulation codes, we begin our process by viewing the visual differences between simulations through comparative visualization. Spreadsheets/small multiples are a common method to present visual differences [6] and have been adopted in many visualization tools.

Recently, researchers started to address the potential "change blindness" with using side-by-side comparative visualizations. One of the ways of tackling this is through visualizing results from different simulations in the same frame or volume [4]. While we attempted to use a visual differencing scheme similar to Haroz et al. [4], there were significant differences between simulation codes that resulted in visually cluttered results. Additionally, our case study scientists preferred side-by-side comparisons found in visualization spreadsheets. Though, we recognize that there needs to be more study of the effectiveness of different comparative visualization methods.

The direct visualization large simulations, minus comparative visualization component, also introduces visual clutter, making important structures not easily discernible in the resulting images. To address this problem, we use feature-based approaches to extract meaningful higher-level structure from scientific data sets. A feature is a defined region in the data set that satisfies constraints, and the exact definition of a feature varies by the scientific domain. In general, a feature describes a coherent structure, or an "effect", that persists for a "significant" amount of time [10]. Using feature-based visualization and analysis allows us to reduce clutter and to quantify and classify simulations.

Quantitative visualization allows us to extract value-based information from the data, through such tools like charts and plots, to numerically describe the differences in simulations. Peskin et al. advocates for interactive quantitative visualization [9] for enhancing scientific and engineering computational simulation prototyping. Recent efforts in interactive quantitative visualization include query-driven approaches [3] and quantitative visualization programming [8].

5 Verification of Simulation Codes

Our case studies, in Sections 7 and 8, describe the specific motivations for the importance of code verification in their respective scientific fields. Under this assumption that code verification is important, we describe our feature-based, visualization-assisted process for verification of simulation codes.

- **Registration of the codes:** The simulation results are registered making sure all measurable axes, such as time steps and spatial coordinates, are in agreement.
- **Step 1: Define (or refine) features:** A measurable feature is defined and a measurement of the feature is defined, as well. A classification algorithm is used to identify the feature in the data to be compared. This may be as simple as a density metric or as complex as a precise structural feature. The feature extraction is applied to all of the simulation data so that the simulations can be visualized and measured in the same manner. If this is a additional pass through the process, a feature can be redefined or refined based on results of hypothesis testing in Steps 3 and 4.
- **Step 2: Formulate (or refine) a hypothesis about the measurable feature in the simulation codes:** Given the feature and measurement defined in Step 1, we make a hypothesis about the feature in the simulation codes. If

this is a additional pass through the process, the hypothesis may be refined based on previous hypothesis testing from Steps 3 and 4.

- **Step 3: Qualitative comparative visualization:**[1] Visualizations created in this step are compared by 2D or 3D rendered features that provide intuition of the differences between the simulations. This step requires visualization and domain expertise to effectively identify the simulation differences. The main point of this step is to provide intuition of the similarities or differences between codes that may drive future queries.
- **Step 4: Quantitative comparative visualization:** We use plots and charts to visualize the feature measurement, defined in Step 1, which provides the quantitative, numerical differences between the simulations. The quantitative results allow us to precisely test the hypothesis from Step 2.
- **Repeat starting at step 1 until the codes are verified:**

Our verification process has a well defined, iterative structure that can be procedurally followed, step-by-step. As the scientific method demands evidence to support a given hypothesis, our process contains the generation of the evidence as key steps. To show the validity of our method, we provide the following case studies as evidence, the application, and effectiveness of our process for code verification.

6 Sidebar: Comparative Visualization Spreadsheet Creation

We discuss some of the details of how we implemented our side-by-side comparative visualizations in ParaView (*www.paraview.org*). Through ParaView, we are able to automatically generate a series collection of related side-by-side visualizations in a one or two dimensional visualization spreadsheet. This is done by varying the parameters of the visualization across the axes of the spreadsheet through the comparative view inspector. The view inspector, shown in Figure 1, allows us to configure how one or more parameters of a visualization vary along the cells of the comparative view. By adding a parameter, values for each cell in the comparative visualization can be set in the accompanying parameter spreadsheet.

Figure 2 is an example of two comparative views, side-by-side, generated for cosmology verification code. The left three columns are a two dimensional comparative view in which the simulation type (a parameter of the reader) varies horizontally and the halo size threshold (a parameter of the threshold filter on halo mass) varies vertically. The right column is a one dimensional comparative view of the halo counts over time which varies by the a halo size threshold.

One of the challenges we faced was how to use ParaView's comparative visualization capabilities to compare data from different simulations. The comparative view was designed to vary the parameters of pipeline objects and could not easily handle our use case, which varied files from different simulations. The solution we devised was a "meta reader" that is aware of a collection of data sets. This reader has an additional parameter which is the index of the data set, corresponding to the simulation. By varying this parameter, we were able to vary the simulations through the standard comparative view interface.

To generate each of the cells in the comparative visualization, we utilize ParaView's keyframe animation capability under the covers. We treat each cell as a keyframe and iterate over the keyframes, modifying the parameter values. Once the pipeline is run for a keyframe, we cache the visual representation. The memory consumption for a comparative view is equal to that used by pipeline execution of one cell plus the memory used by of all cached representations in the comparative view. Due to the demand-driven execution of VTK, it is possible that the pipeline execution time can be lower, on average. Furthermore, the comparative spreadsheet view can be displayed on a tiled display, leveraging ParaView's parallel distributed rendering engine.

[1] We use the term *qualitative* to describe 2D and 3D representations, such as isosurfaces and volume rendering, and the term *quantitative* to describe numerical representations, such as spreadsheets and plots. Clearly, the former visualizations represent quantitative data, however, we use this terminology to distinguish between how the former visualizations are used for exploration and intuition, and the latter is used for measurements and analysis.

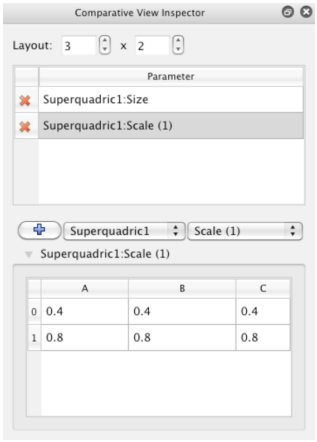

Figure 1: The Comparative View Inspector panel in ParaView to set the varying visualization parameters.

7 Case Study: Verification of Cosmological Simulations

Verifying the accuracy of cosmological simulations is challenging due to the multi-scale complexity in time and space. The spatial dynamics range of realistic problems can vary between two to five orders of magnitude. In our study, we designed an experiment that demands high resolution to verify whether cosmological simulations reach the desired accuracy for gravitational interactions.

We studied three cosmological simulations: GADGET-2, MC^2, and Enzo. Each of these codes were independently developed by different groups around the world; GADGET-2 at the Max Planck Institute, MC^2 at Los Alamos National Laboratory, and Enzo at University of California San Diego. While all three simulations are solving the same N-body problem, the evolution of the dark matter distribution in an expanding universe, their underlying algorithms are different. GADGET-2 is a tree particle mesh (tree-PM) code, MC^2 is a pure particle mesh (PM) code, and Enzo is adaptive mesh refinement code (AMR). All three codes are well-established and are used extensively for cosmological simulations.

With the same initial conditions, each simulation should generate similar results. To investigate this, we ran simulations of the Standard Model of cosmology in a box of 90.14 Megaparsec (Mpc) in each spatial dimension, with 256^3 particles on each simulation and 100 time steps. The force resolution of in a simulation box this small is, in principle, sufficient to capture properties of individual features. The three codes we chose represent the different flavors of algorithms for N-body simulations. Representing AMR simulations, Enzo was run at a 256^3 base mesh and two levels of refinement leading to a 1024^3 mesh peak resolution. MC^2, representing pure PM codes, was run with a 1024^3 mesh, and representing high-resolution tree-PM code, we used GADGET-2.

7.1 Comparison of the Evolution of Halo Population

We demonstrate how the feature-based, visualization-assisted process can be used in a scientific setting to answer specific questions in an efficient manner. In the following example, we study the evolution of the halo population in the cosmological simulations over time. This study addresses differences in statistical measures from the simulations, such as halo counts as function of mass and time, and therefore global results of the simulations. We describe our verification of the cosmology codes to highlight how the process led us to key scientific results. We consider our results and ease that the cosmologists were able to verify the simulations to be one of the best tests of the effectiveness of our structured process.

Assuming we have registered the codes we wish to study using the same initial conditions, Step 1 in our process

is to define a feature we wish to measure. In cosmology, the halo is a central feature to the analysis of dark-matter simulations. In lay terms, a halo is a cluster of particles in a cosmological simulation. Unfortunately there is no unique definition for identifying a dark matter halo. In our study, we exclusively used the Friends-of-Friends (FOF) metric [2] for halo definition. In this definition, a pair of particles are designated as friends if their distance in space is within a certain threshold termed the *linking length*. A halo is defined as a set of particles connected by one or more friendship relations, i.e., friends-of-friends.

The number of halos as a function of their mass is called the *halo mass function* or *mass function*. The mass function and the time evolution of it describes much about the formation of cosmological structures. The precise measurement of the mass function from observations, and its prediction from theory, are a field of very active research in cosmology. We use this metric as the measurement by which we will compare the simulations, completing step 1 of our process.

A popular N-body algorithm for is the adaptive mesh refinement (AMR) method, represented by Enzo. In AMR codes, the simulation begins with a uniform base grid and refines specific regions in the simulation to higher grid resolution as refinement criteria are met. Previously, AMR methods have been successfully used in the simulation of single objects, e.g., supernovae. In cosmological simulations, the simulation starts with a smooth Gaussian density field, and over time higher density regions evolve. The grid refinement criterion is usually dictated by a density threshold: if a certain density is reached, the grid is refined. We refer to the highest level of refinement as the *peak resolution*.

AMR methods appear to be a natural way to perform cosmological simulations. The intuition is that higher resolutions are only needed in high density regions, and at later simulation time steps. Therefore, using AMR reduces the simulation hardware requirements and saves computational time. Though, the benefits of AMR methods need to be tested and their results verified, which leads to our study and the formulation of the hypothesis in Step 2 of our process:

Hypothesis 1: An AMR code with a peak resolution equivalent to a uniform grid code should resolve all halos of interest.

In Step 3 of our process, we first test our hypothesis by performing a qualitative visualization comparison of the feature of interest. We visualized the halos from the three simulations, at the last time step, shown in Figure 2. In this case, the visual interpretation of the results is easy: our hypothesis is false. Visually, Enzo has far fewer halos.

Following up with Step 4 of our process, the quantitative visualization or analysis step, we find that Enzo has 29099 halos, MC^2 49293 halos, and GADGET-2 has 55512 halos. One could argue that the pure halo count would have been enough to falsify our hypothesis. Though, the Step 3 visualization of the spatial distribution of the halos contains much more information than a pure number though, and serves to provide further intuition to refine hypotheses and visualizations. For example, Figure 2 hints that the halo count is different in the entire simulation volume and not just in specific regions where the AMR code might not have refined to the highest level of resolution.

Discussion

- We incorporate the feature extraction code within the visualization tool, instead of using the halo finding code in the simulations. In practice, simulations may have their own analysis methods which makes comparisons of the simulations difficult if we used these metrics. By unifying the feature extraction by providing it in the visualization tool, we remove any measurement ambiguities between different analysis codes. Additionally, there is a significant time lag in the visual analysis workflow if we needed to re-run each simulation to produce features for the comparative visual analysis Steps 3 and 4.

- To produce the comparative visualizations, we could generate them by hand by varying the parameters in the visualization tool. For example to generate Figure 2, we would need to input the files for each different simulation and regenerate the same visualization pipeline to generate separate visualization windows. Managing these parameter changes and windows by hand is tedious task, and is prone to error due to manual repetition. In practice, our process supports the automatic generation and management of comparative visualizations (described in the Sidebar 6), and this automation reduces the time to complete Step 3 from hours to minutes.

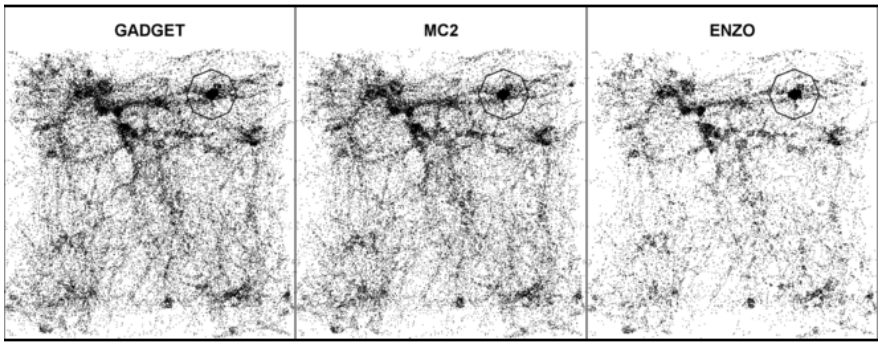

Figure 2: Comparative visualization of the distribution of the halos for the simulation results in the final time step. Each halo is represented by a point. It is obvious from these images that Enzo has fewer halos throughout the simulation volume, and therefore our Hypothesis 1 is false.

- An alternative is to rely on purely analytical methods, skipping Step 3 in our process. As previously mentioned, we require visual inspection of the data as we find that this provides additional insight (in this case, the data's spatial distribution) that drives the verification process forward quicker than a pure analytical approach. The intuition provided by Step 3 is a key part of the process to refine visualizations and hypotheses.

Repeating back to Step 1, we aim to refine our hypothesis in such a way that we gain a clear understanding why Hypothesis 1 was false. We focus on an additional feature in the simulation: the halo mass. This feature or measurement is provided by the halo finder and is important to understand our results in more detail.

In [5], a criterion was derived for the force resolution required to resolve halos of a certain size. The force resolution of Enzo's base grid allows it to capture halos with more than 2500 particles, the first refinement level allows it to resolve halos with more than 300 particles, and the highest refinement grid allows the resolution of halos to be as small as 40 particles. Since MC^2 uniform base resolution is equivalent to Enzo's peak resolution, MC^2 should reliably capture halos with at least 40 particles as well. While halos with fewer particles are often unphysical (random particles which are by chance close to each other and are linked together even though they do not form a bound structure), we investigate the halos with less than 40 particles.

In Step 2, we refine our hypotheses to form two possible, related explanations for the deficit of halos:

Hypothesis 2a: The halos do not form at early times when the base resolution is still very low and cannot be recovered later.

Hypothesis 2b: Only halos of a certain size – dictated by the base grid and *not* by the peak resolution – can be captured correctly.

We test our two new hypotheses in Steps 3 and 4 by visualizing and (i) dividing the halos into mass bins, and (ii) tracking the number of halos forming in the simulation over time in the separate mass bins.

The results of our comparative qualitative and quantitative visualization analysis testing 2a and 2b, Step 3 and 4 in our process, are summarized in Figure 3. We would like to emphasize that this figure only shows the final result of an interactive, iterative process.

From top to bottom, Figure 3 shows the different mass ranges, starting with the lightest halos and ending with the heaviest halos. The first three columns show the halo results for the different simulations at the final time step, and column four shows the quantitative results of halo count in mass bins over time. The qualitative visualization (columns 1 – 3) of the simulation results does not lead to an immediate answer as it did in the previous case for Hypothesis 1. Though, it appears visually that Enzo has fewer light halos and fewer medium halos. A judgment about the number of heavy and extra-heavy halos by inspection of the qualitative plots is not possible and therefore Step 4 in our process becomes essential.

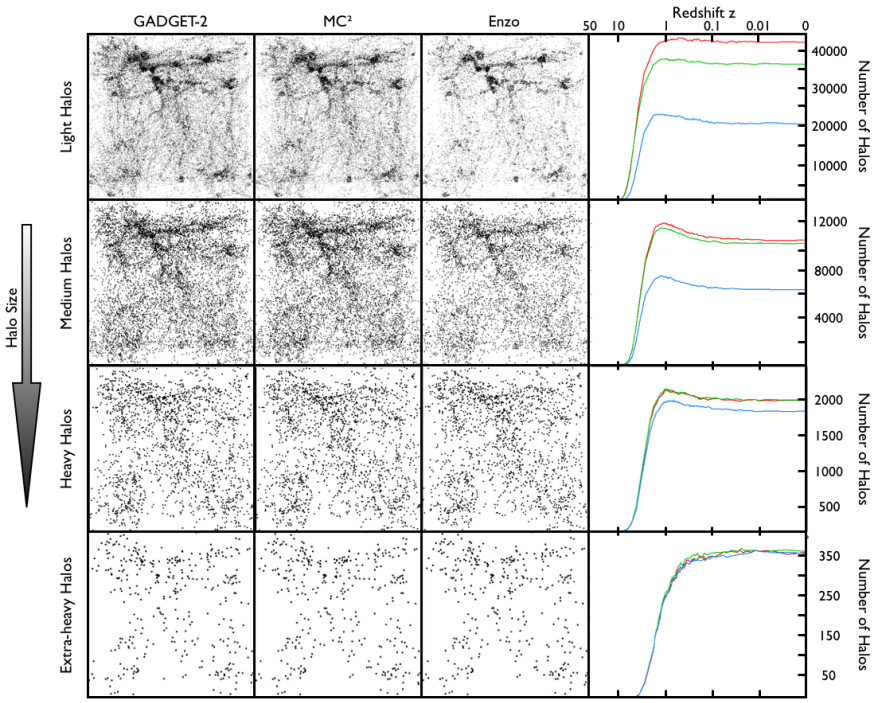

Figure 3: First three columns: comparative qualitative visualization of the halos in different mass ranges (light to extra-heavy from top to bottom) in the final time step. We present results from GADGET-2 in red, from MC^2 in green, and from Enzo in blue. A light halo contains 10-40 particles, a medium halo contains 41-300 particles, a heavy halo contains 301-2500 and an extra-heavy halo contains more than 2500 particles. Fourth column: comparative quantitative visualization of the number of halos in different mass ranges over the time range. The combination of both allow the assessment of the validity of hypotheses 2a and 2b.

Column 4 in Figure 3 shows the results displayed in a quantitative plot over time. We show the number of halos and their evolution over time in four different mass bins. Now it is apparent that Enzo (blue curve) has a large deficit in halos for the light and medium halos, and it also lacks heavy halos. Only the very heavy halos are captured by Enzo correctly. We therefore have confirmed that our hypothesis 2b is supported. Figure 3 also contains information about Hypothesis 2a: it is apparent that the deficit of halos in the Enzo simulation is rooted in early time steps of the simulation. Halos are missing from the beginning in Enzo, and are not recovered over time, e.g., the curve for the halos count does not catch up with the other simulations at later time steps.

At this point, the verification process has been successfully completed for AMR codes: we now have an understanding of what halos can be captured reliably by AMR codes. With this verification, we are armed with the knowledge of how to improve the performance of AMR codes, with respect to halo resolution. They can can be improved by either starting the simulation with a much larger base grid, or by setting the refinement criteria to be much more aggressive, such that the refinement starts long before the first halos form.

Discussion

The cosmologists found our structured process useful since it enforced quantitative evidence gathering for the scientific process, and it also guided their next exploratory steps in the massive search space of possibilities.

- Our visualization process structures the analysis so that it produces a chain of documentary evidence that supports the verification and scientific process. It also generates a document for reproducibility and dissemination.

The hypothesis text and Figures 2 and 3 were used to present the AMR issues to the Enzo scientific team.

- In the past, exploratory visualization and quantitative analysis were typically separate tasks. Analysis codes, such as halo identification and counting, are not typically integrated within the visualization process and vice-versa. By following our process, the qualitative visualizations are used to gain insight, leading to the correct questions to ask, and leading to quantitative plots. In summary, the process allowed the cosmologists to be more productive in their code verification work.

- Our process requires iterative, interactive exploration. It is hard to convey in print the "search and exploration" aspect of our process: (i) the work involved to identify a qualitative and quantitative measure that would distinguish a measurable difference between the simulations, and (ii) the refinement of the visualizations and hypotheses. Alternative approaches that do not include this structured visualization might have missed the key aspects of the results (formation issues based on density regions) that we were able to find through our iterative refinement process.

8 Case Study: Verification of Ocean Models

Modern global climate models (GCMs) are composed of numerous components, including atmosphere, ocean, land cover, sea ice, and land ice. These components communicate surface fluxes, such as heat, through a coupler at regular intervals. GCMs are an important tool for scientists to study scenarios of climate change. Along with historical observations, paleoclimate records, and theoretical understanding of radiative processes (energy transfer), GCMs form part of the growing body of evidence that increasing greenhouse gasses warm the atmosphere. GCMs offer the special advantage that scientists and policy makers can explore many possible future scenarios to see what may be in store, and to potentially add feedback to debates on policies to curb greenhouse gas emissions.

To verify the correct operation of a new prototype ocean-climate code, we compare it against existing simulations. Our standard for comparison is POP, the Parallel Ocean Program [11], developed and maintained at Los Alamos National Laboratory. After 15 years of use, POP has become trusted by the climate modeling community. Since a new prototype model is a completely new ocean simulation, verification by comparison to POP is an important step in the development process.

POP runs on a logically rectangular (for example, latitude-longitude) horizontal grid. Our new prototype code has a horizontally unstructured grid that can support rectangles like POP, as well as icosahedral (hexagonal) grids. A strong motivation for the development of this particular ocean model is its ability to use variable density meshes, where high resolution regions can be immersed in a low-resolution global grid with spacing varying continuously between the two grid densities. This feature can be used to study regional climates, to add resolution to areas with strong jets and eddies, or to run high-resolution coastal simulations.

These differences mean that even with all other parameters and initial conditions set identically, code verification tests should match qualitatively, but bit-for-bit matching is not possible. In the verification process, the challenge is to determine how close of a match is sufficient to approve the new prototype code, to expose coding errors in the new model, and to understand valid differences between the two models.

8.1 Verifying a New Ocean Model

In the following, we describe our iterative verification process applied to verifying a new prototype ocean model by comparing it against POP. To find potential development errors, we compare the two codes with conditions matching as closely as possible, first in the simplest setting, and in the future, adding complexity in a step-wise manner. The initial tests described here are the simplest in this process: the horizontal grid is rectangular and identical for POP and the prototype; vertical grid depths are identical; all parameters, such as viscosity and diffusion coefficients, are spatially constant; the surface wind forcing is an idealized cosine function of latitude; and initial temperature and

salinity (and resulting density) profiles are simple linear functions of depth (density varies linearly from 1002 to 1016 kg/m^3).

There are two particular items that differ between POP and the prototype in these comparisons and both are fundamental to the models and thus cannot be changed. First, POP computes velocities at the corners of each cell, while the prototype code computes velocity at the cell edge. Second, the prototype code solves a different formulation of the momentum equation than POP. Finally, simulations of the POP ocean model and the prototype have different units (cgs and mks units, respectively) so care must be taken in comparing output.

For Step 1 of the code verification process, we choose the sea surface height (SSH) as the measurable feature. Global SSH is rich in information and is often the first variable analyzed by ocean modelers. The world's oceans are, to first order, in geostrophic balance, which means that pressure gradients and Coriolis forces are balanced and SSH contours are streamlines for the vertically uniform (barotropic) flow. In other words, a contour plot of global SSH also communicates the large scale flow. Thus, comparisons of SSH from two different models will reveal differences in their overall dynamics.

This leads us to the formulation of our hypothesis in Step 2:

Hypothesis 1: The global 2D SSH of POP and the new prototype code should match.

By "match", we mean that the SSH results from the two models should be qualitatively similar in their spatial structures, and that quantitative values should match within 10% after a few hundred simulated days. Step 3 of our process (Figures 4 and 5) shows that the SSH of the two models display general agreement after 150 days and six years. This agreement is most prominent in the semi-circular pattern of SSH in each ocean basin.

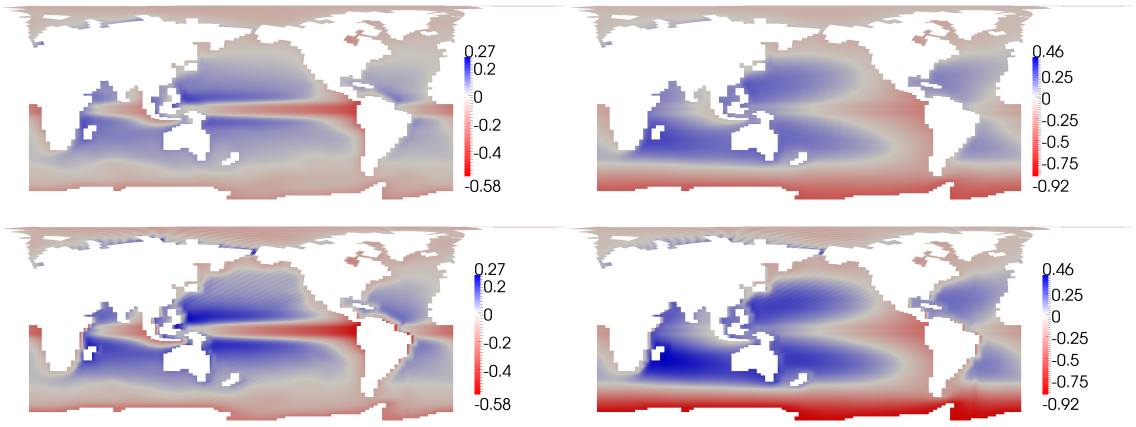

Figure 4: Sea surface height for the prototype code (top) and POP (bottom) after 150 days and six years (left to right). Images show general agreement in the large-scale flow supporting Hypothesis 1.

Additionally, we see that there are grid-scale oscillations in POP that appear as diagonal ripples or banding. POP also has large SSH anomalies on some boundary cells, particularly along the western edge of ocean basins. Both of these issues do not appear in SSH plots from the prototype model, and are notable improvements over POP. Both issues can be seen in more detail in Figure 5, showing the magnitude of differences that may not be readily apparent in the earlier SSH images. For Step 4, we use histograms of SSH (Figure 6) that show there is a wider spread for the POP data due to the boundary cell anomalies. This process has helped the developers identify and understand the underlying issues that cause disagreements between the two models.

From Figure 5, it is reveled after six years of simulation time there is a significant difference between POP and the prototype model. The SSH drops more strongly across the Southern Ocean (towards Antarctica) in POP than in the prototype model. This indicates that the Southern Ocean's circulation is stronger in POP than in the prototype,

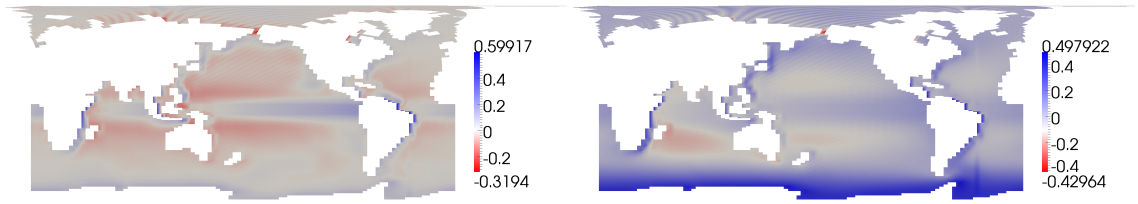

Figure 5: Sea surface height difference for the prototype code and POP after 150 days and six years (left to right). The images show fine-grain differences between the two models that may not be apparent in Figure 4, leading to new questions.

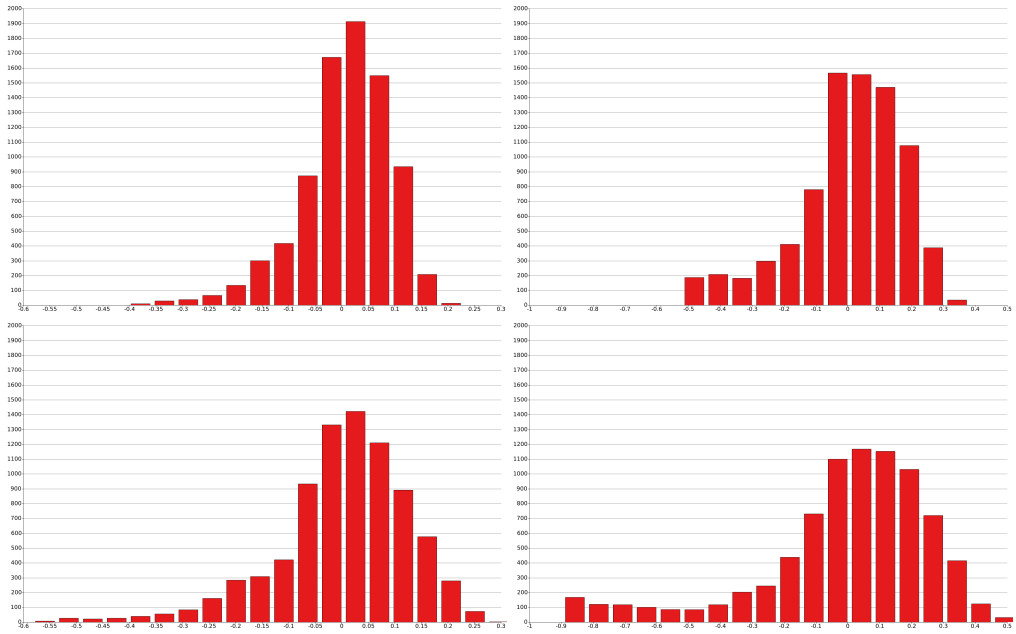

Figure 6: Histogram of sea surface height for the prototype code (top) and POP (bottom) after 150 days and six years (left to right). POP has a wider spread due to boundary cell anomalies.

and this difference is reinforced by the dark blue in the Southern Ocean from the difference plot (right side of Figure 5). The histogram of SSH after six years (right side of Figure 6) also shows this behavior in POP's negatively skewed distribution.

This opens up new questions, and starts our process anew. To understand the difference in SSH values between POP and the prototype code, we refine our visualization and metric (Step 1) to evaluate additional variables. The relationship between the gradient of SSH and depth averaged (barotropic) velocities is well understood for geostrophic (pressure gradients and Coriolis forces are balanced) flows. Zonal (eastward) velocity is proportional to latitudinal gradients of SSH. The flows in these test simulations are expected to be nearly geostrophic (for example, ageostrophic forces include surface winds and bottom drag). Therefore, the zonal (eastward) and meridional (northward) velocities provide additional information on the dynamics of the ocean models, which can help explain the SSH differences in the models.

This leads to our new hypothesis in Step 2:

Hypothesis 2: Depth averaged (barotropic) velocities in the Southern Ocean of POP are larger where there are larger gradients in SSH.

For Step 3 of our process, we use 2D plots of velocity from the top layer of the ocean model (Figures 7 – 9). In both models, zonal velocities show a strong westward jet at the equator, banded by eastward currents to the north and south (Figure 7). This is a direct response to the idealized wind forcing applied at the surface. Also, meridional velocities are poleward on either side of the equator (Figure 8). This is a secondary effect of the wind forcing: the Coriolis force turns the equatorial jet to the right in the Northern Hemisphere and to the left in the Southern Hemisphere. The strong similarity between POP and the prototype code in these plots, reinforced by the difference plot in Figure 9, shows that the wind forcing, advection, and Coriolis terms are working correctly in the new ocean model.

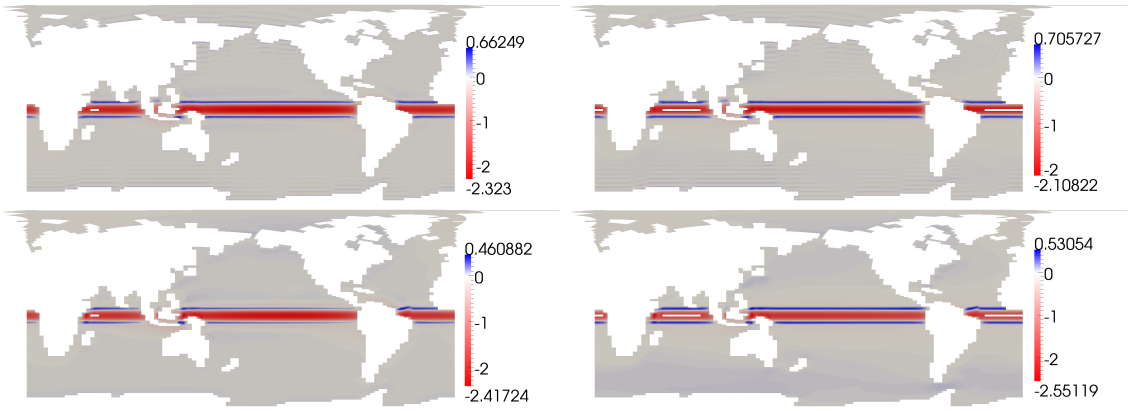

Figure 7: Zonal (eastward) velocities for the top layer of the prototype code (top) and POP (bottom) after 150 days and six years (left to right). Wind forcing, advection, and Coriolis terms appear to work correctly in the new model.

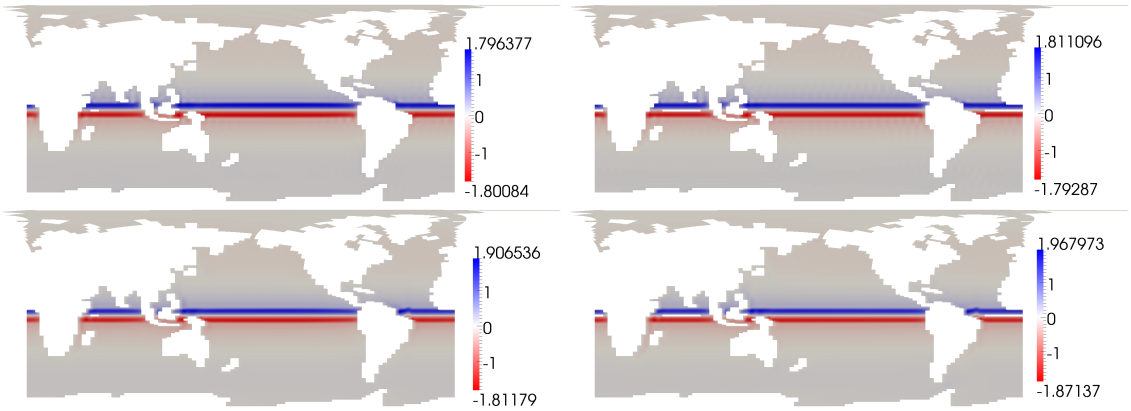

Figure 8: Meridional (northward) velocities for the top layer of the prototype code (top) and POP (bottom) after 150 days and six years (left to right). Wind forcing, advection, and Coriolis terms appear to work correctly in the new model.

The 2D plots of velocity do not support Hypothesis 2 or help us understand the SSH difference in the Southern Ocean because the velocities are weak compared to the equatorial currents. To circumvent this, using Step 4 allows

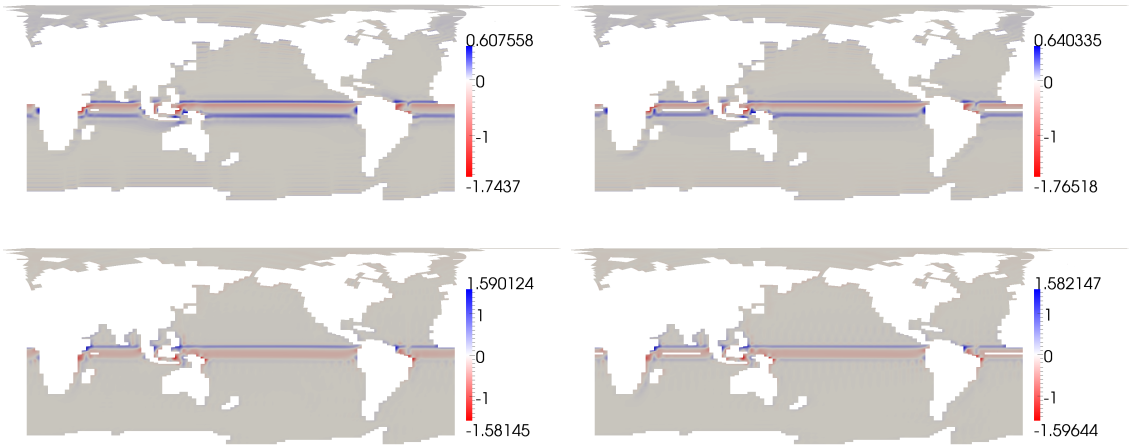

Figure 9: The difference between the prototype code and POP velocities. Zonal (eastward) velocities are on top and meridional (northward) velocities are on the bottom, both after 150 days and six years (left to right). There appears to be little difference between the models, but this figure and Figures 7 and 8 do little to support Hypothesis 2.

us to view the velocities associated with the SSH gradient in vertically averaged statistics (Figure 10). We see that the velocities (y-axis) match reasonably well between POP and the prototype code after 150 days. Velocities above 80 north in the Arctic (right-side of the plot on the x-axis) suffer from small sample sizes. After six years, the meridional velocities still agree, but zonal velocities differ by a factor of two in the Southern Ocean (between 60 and 70 south latitude on the x-axis) in our quantitative measurements. This confirms the difference in SSH observed previously, supports Hypothesis 2, and indicates that further future investigation is required to explain differences between the two codes in this regard.

9 Conclusion

We have described a visualization-assisted process for the verification of simulation codes. The need for code verification stems from the requirement for very accurate models to interpret or predict observational data confidently. We find that combining comparative, feature, and quantitative visualization together greatly improves the efficiency and insight in performing code verification. Furthermore, our process provides a step-by-step formula which generates a document trail for reproducibility.

We presented two case studies: one that verified the accuracy of cosmological AMR N-body simulation methods, and the other tested new ocean models, both through our iterative visualization process. Specifically for cosmology, we gained insight to the appropriate parameter settings for AMR cosmology codes. As AMR is a common optimization strategy in many scientific simulations, our verification process can provide guidance for similar projects in other fields. In ocean model development, the iterative visualization process serves as a tool to validate a newly created code. This process improves our ability to analyze data for understanding and quantifying model differences and to pursue coding errors that cause unexpected differences in models.

References

[1] M. Chen, D. Ebert, H. Hagen, R. Laramee, R. van Liere, K.-L. Ma, W. Ribarsky, G. Scheuermann, and D. Silver. Data, information, and knowledge in visualization. *IEEE Computer Graphics and Applications*, 29(1):12–19,

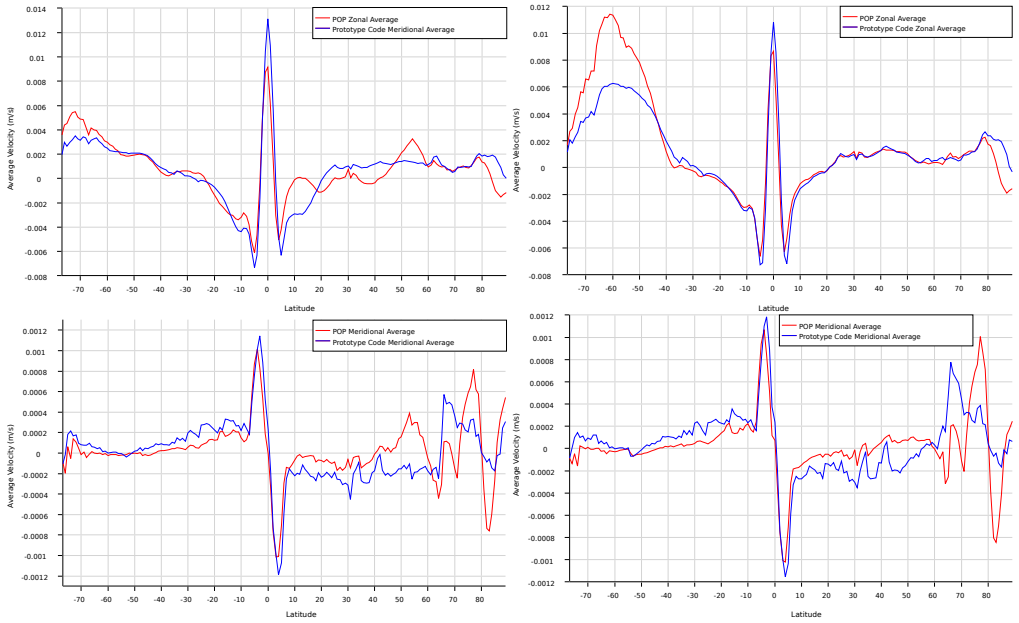

Figure 10: Zonal (eastward) velocity (top) and meridional (northward) velocity (bottom) after 150 and six years (left to right). POP is the red curve, while the new prototype is the blue curve. Larger zonal velocities in POP in the Southern Ocean (the left side of the upper right figure) agree with earlier SSH images in Figure 5, highlighting the SSH difference between the codes and supporting Hypothesis 2.

Jan–Feb 2009.

[2] M. Davis, G. Efstathiou, C. Frenk, and S. White. The evolution of large-scale structure in a universe dominated by cold dark matter. *The Astrophysical Journal*, 292:371–394, May 1985.

[3] L. Gosink, J. Anderson, E. Bethel, and K. Joy. Query-driven visualization of time-varying adaptive mesh refinement data. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1715–1722, Nov/Dec 2008.

[4] S. Haroz, K.-L. Ma, and K. Heitmann. Multiple uncertainties in time-variant cosmological particle data. In *IEEE Pacific Visualization Symposium*, pages 207–214, Mar. 2008.

[5] K. Heitmann, Z. Lukić, S. Habib, and P. Ricker. Capturing halos at high redshifts. *The Astrophysical Journal*, 642:L85–L88, May 2006.

[6] T. Jankun-Kelly and K.-L. Ma. Visualization exploration and encapsulation via a spreadsheet-like interface. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):275–287, July 2001.

[7] J. Kehrer, F. Ladstädter, P. Muigg, H. Doleisch, A. Steiner, and H. Hauser. Hypothesis generation in climate research with interactive visual data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1579–1586, Nov/Dec 2008.

[8] P. McCormick, E. Anderson, S. Martin, C. Brownlee, J. Inman, M. Maltrud, M. Kim, J. Ahrens, and L. Nau. Quantitatively driven visualization and analysis on emerging architectures. *Journal of Physics: Conference Series*, 125:012095 (10pp), 2008.

[9] R. Peskin, S. Walther, A. Froncioni, and T. Boubez. Interactive quantitative visualization. *IBM Journal of Research and Development*, 35(1–2):205–226, Jan/Mar 1991.

[10] D. Silver and X. Wang. Tracking and visualizing turbulent 3D features. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):129–141, Apr-Jun 1997.

[11] R. D. Smith, J. K. Dukowicz, and R. C. Malone. Parallel ocean general circulation modeling. *Physica D*, 60:38–61, 1992.

[12] S. Solomon, D. Qin, M. Manning, Z. Chen, M. Marquis, K. Averyt, M. Tignor, and H. Miller, editors. *Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change, 2007*. Cambridge University Press, United Kingdom and New York, NY, USA, 2007.